molecules

MDPI

Article

Reducing Ensembles of Protein Tertiary Structures Generated De Novo via Clustering

Ahmed Bin Zaman [1,†], Parastoo Kamranfar [1,†], Carlotta Domeniconi [1,*,†] and Amarda Shehu [1,2,3,4,*,†]

[1] Department of Computer Science, George Mason University, Fairfax, VA 22030, USA; azaman6@gmu.edu (A.B.Z.); pkamranf@gmu.edu (P.K.)
[2] Center for Advancing Human-Machine Partnerships, George Mason University, Fairfax, VA 22030, USA
[3] Department of Bioengineering, George Mason University, Fairfax, VA 22030, USA
[4] School of Systems Biology, George Mason University, Fairfax, VA 22030, USA
* Correspondence: cdomenic@gmu.edu (C.D.); amarda@gmu.edu (A.S.); Tel.: +1-703-993-1697 (C.D.); +1-703-993-4135 (A.S.)
† Current address: MS 4A5, 4400 University Dr., Fairfax, VA 22030, USA.

Received: 14 January 2020; Accepted: 28 April 2020; Published: 9 May 2020

Abstract: Controlling the quality of tertiary structures computed for a protein molecule remains a central challenge in de-novo protein structure prediction. The rule of thumb is to generate as many structures as can be afforded, effectively acknowledging that having more structures increases the likelihood that some will reside near the sought biologically-active structure. A major drawback with this approach is that computing a large number of structures imposes time and space costs. In this paper, we propose a novel clustering-based approach which we demonstrate to significantly reduce an ensemble of generated structures without sacrificing quality. Evaluations are related on both benchmark and CASP target proteins. Structure ensembles subjected to the proposed approach and the source code of the proposed approach are publicly-available at the links provided in Section 1.

Keywords: protein structure prediction; tertiary structure; decoy ensemble; reduction; clustering

1. Introduction

The three-dimensional/tertiary structure of a protein molecule is key to determining its array of activities in the cell, as proteins employ their tertiary structures to interface with other molecular partners [1]. Though the path to decoding protein function seems to go through its tertiary structure, determining the biologically-active structure(s) of a protein poses many challenges in both the wet and dry laboratory [2]. Great progress has been made in wet-laboratory structure determination, but these advancements nonetheless lag behind the technological advances in gene sequencing; the increasingly faster and cheaper high-throughput gene sequencing technologies have yielded millions of protein sequences [3]. In contrast, the number of known biologically-active/native protein structures is an order of magnitude less. For instance, as of April 2020, the number of experimentally-known structures deposited in the Protein Data Bank (PDB) [4] is around 160,000.

The above discrepancy continues to motivate protein structure prediction (PSP) in dry laboratories. One of the most challenging settings is that of de-novo/template-free PSP, where the only direct information about a target protein at hand is its amino-acid sequence [5]. This is typically the case for proteins that do not have other, sufficiently-similar protein sequences with known structures that could otherwise serve as structural templates upon which to thread the target sequence [5]. It is worth noting that there is some evidence of stagnation in the rate of discovery of new protein folds [6]. This suggests that reasonable structural templates may soon be found for increasingly many more target proteins and transferred onto them via homology modeling.

Prediction in a de-novo setting is carried out in two stages. In the first stage, the focus is on computing or generating physically-realistic structures. Structure generation algorithms address an optimization problem; they seek tertiary structures that minimize the interaction energy among the atoms of a given target protein. It is now well-known that the energy functions designed in computational laboratories are inherently inaccurate [7–9]. In particular, one cannot infer that a lower-energy structure is more similar to the sought native structure than a higher-energy structure. In a blind setting, it is unknown which structures are sufficiently close to the sought native structure and can be deemed near-native. The objective of the second stage is to determine such structures. The algorithms put forth for this purpose are known as model assessment, model selection, or decoy selection; the terms "model" and "decoy" refer to a computed/generated structure in this context.

Great algorithmic advances have been made in structure generation, most remarkably by Rosetta [10], Quark [11], and others [12–16]. Recent works have investigated incorporating complementary information like sequence-predicted contacts and constructing new energy functions based on predicted contacts or distances of pairs of amino acids for structure generation [17–19]. Algorithms utilizing deep neural networks are also becoming increasingly popular [20–24]. Such advances are documented in the Critical Assessment of protein Structure Prediction (CASP), which is a biennial community experiment/competition that assesses progress in PSP in several categories, including the template-free category (also referred to as free modeling) [25].

Outstanding challenges concern how to control the quality of generated structures. Despite clever choices in representing tertiary structures [5,26], the search space that has to be explored by a structure generation algorithm is vast and high-dimensional. Consider the following back-of-the-envelope calculation: on a protein not exceeding 100 amino acids, modeling three backbone dihedral angles per amino acids results in around 300 variables; these constitute the 300 dimensions of the search space. In addition, the energy function that structure generation algorithms optimize (by seeking structures that reside in local minima of the energy surface) is noisy.

Currently, when employing popular platforms, such as Rosetta and Quark, the recommendation from developers is to generate as many structures that can be afforded, with actual numbers practically varying between 10,000 and 20,000 structures. This recommendation acknowledges that more structures may translate into higher likelihood that some will reside near the sought native structure. The recommendation is impractical for various reasons. While generating structures used to be significantly more expensive than analyzing them, now this relationship is less imbalanced. Great progress in software and hardware have made it less costly to generate structures. For instance, algorithms operating under the umbrella of evolutionary computation can generate hundreds of thousands of structures [12,13,16,17]. Algorithms tasked with analyzing these structures now may have to additionally deal with a data size issue.

Moreover, in most template-free PSP protocols, the end of the first stage adds back the side-chain atoms on each structure and carries out local improvements on the resulting all-atom structures prior to handing them off to the second stage. Adding atomistic detail on a structure is computationally expensive, as the energy function employed has to handle a large number of atoms per structure (that includes all side-chain atoms and all hydrogen atoms for each amino acid). This is exacerbated when the recommendation is to collect large numbers of structures.

In this paper, we propose a novel approach to reduce the size of a structure ensemble. The objective is to do so while retaining the quality of the original ensemble, as measured via several metrics described later in the manuscript in Section 2. To the best of our knowledge, the problem of structure ensemble reduction while preserving quality is unexplored. The problem is also not trivial. In the de-novo setting, it may be tempting to tackle it by discarding higher-energy structures. Indeed, early work in [27] does so before proceeding to cluster the remaining structures for the purpose of model selection. As we show in our evaluation in Section 3, an approach that simply utilizes an energy threshold, which we employ as a baseline for the purpose of comparison, does a poor job at retaining near-native structures. This is not surprising. As the above discussion relates, energy is not a reliable indicator of nativeness.

The proposed approach seeks to retain the inherent organization of the structures in a given ensemble and does so by utilizing clustering algorithms. While this paper evaluates the impact of various clustering algorithms, the contribution is not in proposing clustering algorithms; nor is the focus of this paper in evaluating the vast landscape of existing clustering algorithms. Instead, the contribution of the proposed approach is in leveraging smart featurization of tertiary structures, computationally-expedient clustering in the feature space to capture the inherent organization, and selecting a subset of structures that retain the exposed organization in tandem with additional properties of interest. The latter here are limited to internal energies, but other properties of interest may be utilized that may expand the applicability of the proposed approach to structure ensembles beyond the de-novo setting. The main reason we focus on such ensembles here is due to our ability to generate many structures and the clear need for the proposed approach in de-novo PSP.

It is also worth noting that the efforts described in this manuscript are not targeting the problem of model assessment/selection, though the reduced ensembles may be of use downstream to methods that tackle model assessment/selection. Some of these methods employ clustering, supervised learning, or a combination of learning techniques [27–33]. A preliminary version of this work has appeared in [34]. Here we expand the work by adding clustering algorithms, metrics measuring quality, and detailed evaluations. Rigorous analysis is conducted to determine the optimal settings for the employed clustering algorithms. Evaluations are carried out on diverse target protein datasets that include recent hard, free-modeling domains from CASP12 and CASP13 and show that the proposed approach yields drastic reductions in ensemble size while retaining structure quality.

We note that the proposed approach is publicly-available at https://github.com/psp-codes/reduced-decoy-ensemble (Code doi:10.5281/zenodo.3758031). All structure ensembles subjected to the proposed approach are provided freely on IEEEDataPort at https://ieee-dataport.org/open-access/protein-tertiary-structures-zamanmolecules20 (Data doi:10.21227/gq2v-8k24).

The rest of this paper is organized as follows. We describe the proposed methodology in Section 2. Evaluation is presented in Section 3. The paper concludes with a summary and discussion of future work in Section 5.

2. Materials and Methods

We will refer to a given ensemble of structures as Ω_{gen} and to the reduced ensemble (by our approach) as Ω_{red}. We note that Ω_{red} keeps only a fraction of the structures in the original Ω_{gen}. To efficiently produce a reduced-size structure ensemble Ω_{red} that retains the quality and diversity of the original, full-size structure ensemble Ω_{gen}, the method we propose in this paper leverages fast shape similarity for tertiary structures.

The proposed method consists of three stages. First, a *featurizer* extracts structure features that summarize the three-dimensional shape. Second, the features are then utilized by a clustering algorithm to group structures based on their shape similarity. Third and finally, a *selector* selects structures from each cluster/group identified over Ω_{gen} to populate the reduced ensemble Ω_{red}. In the following, we describe each of these stages in detail. Before doing so, in the interest of clarity, we provide some more details into how the structures we utilize to evaluate our method are generated in the first place.

2.1. Generation of Structures for a Target Protein

Many options are available to generate the ensemble Ω_{gen}, Rosetta, Quark, etc. We choose to utilize an in-house algorithm, the hybrid evolutionary algorithm (HEA) [16], which has been evaluated against Rosetta and other algorithms [14–16]. HEA leverages evolutionary search techniques to balance between exploration and exploitation in a vast search space and has been shown to have higher exploration capability than Rosetta [15,16]. While any structure generation algorithm can be used to generate the Ω_{gen} ensemble for our purposes in this paper, we specifically employ HEA due to its high exploration capability; the algorithm can generate hundreds of thousands of structures for a target protein (given its amino-acid sequence) in a matter of hours.

In its functional core, the HEA builds over the Rosetta engine and utilizes its representation of structures, as well its suite of energy functions. In summary, HEA evolves a fixed-size population of structures over a number of generations. The initial population is constructed by first creating identical extended chains from the amino-acid sequence of a target protein and then randomizing the chains by employing repeated molecular fragment replacements of length 9. The molecular fragment replacement operation in HEA utilizes libraries of fragments of length 9 and 3 generated for a given amino-acid sequence via the Robetta server [10]. The fragments are excised from known native structures of proteins in the PDB. To perform molecular fragment replacement on a structure, first an amino acid index i is chosen at random from the range $[1, L_p - L_f + 1]$, where L_p is the number of amino acids in a given sequence, and L_f is the length of fragment. The fragment composed of amino acids $[i, i + L_f - 1]$ in the given structure is then replaced with a fragment configuration selected at random from the available fragments in the fragment library with the same or similar amino-acid sequence.

In each generation of HEA, the structures in a population are considered parents, and each parent is subjected to molecular fragment replacement of length 3 to produce an offspring. Each offspring is improved by repeatedly applying molecular fragment replacement of length 3. Next, the improved offspring and the top 25% of the parents compete against one another in terms of energy. The top structures survive to become parents for the next generation. Interested readers can learn more about HEA in Ref. [16].

2.2. Stage I: Featurizing Generated Structures

We utilize the Ultrafast Shape Recognition (USR) metrics that were originally introduced in [35] to summarize three-dimensional structures of ligands. USR metrics were used in [35] to expedite searches for similar structures in molecular databases. These metrics have also been used by us and others to expedite robotics-inspired algorithms exploring protein structure spaces for structure [36,37] and motion computation [38,39].

In this paper, we use the USR metrics as features to summarize a tertiary structure. The metrics allow efficient comparison of molecular shapes. They summarize the distributions of distances of all atoms in a tertiary structure from four chosen reference points: the molecular centroid (ctd), the closest atom to ctd (cst), the farthest atom to ctd (fct), and the farthest atom to fct (ftf). Figure 1a shows the locations of these four reference points, with atoms drawn as spheres in a tertiary structure selected for illustration. Once the reference points have been calculated, distances of all atoms from each reference point are calculated next; Figure 1b shows distances of all atoms from one of the reference points, the ctd, by drawing them as lines. The moments of the calculated distance distributions are recorded to summarize a given tertiary structure. Specifically, in our work (as originally in [35]), the resulting distributions are summarized with three momenta, the mean, variance, and skewness. Hence, each tertiary structure in Ω_{gen} is represented by 12 features.

The motivation of encoding each tertiary structure via features is three-fold. First, a lower number of coordinates required to represent each structure reduces the computational time of any algorithm expected to process the generated structures. Second, high data dimensionality has a negative impact on the performance of clustering algorithms [40–42]. Third, unlike representations based on Cartesian coordinates, the USR-based representation is invariant to rigid-body motions (translation and rotation in 3D space).

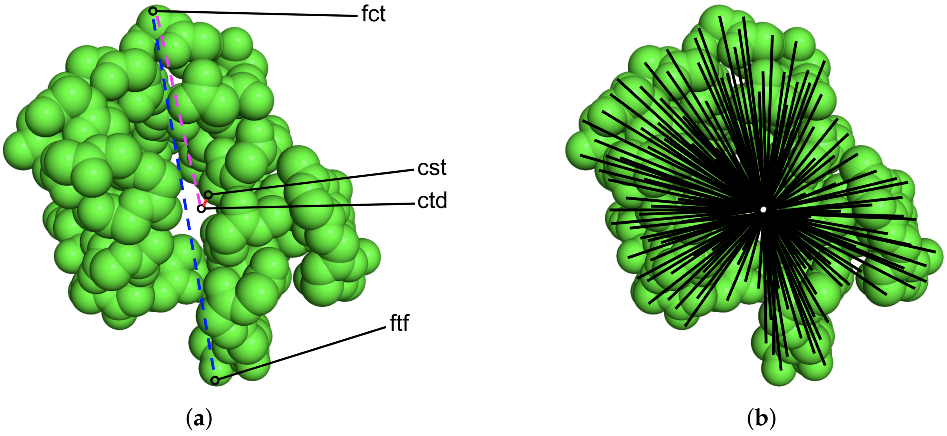

Figure 1. (**a**) The four reference points used to calculate USR metrics on a given tertiary structure are shown here. (**b**) Distances of all atoms from a chosen reference point, the ctd, are also shown.

2.3. Stage II: Clustering Featurized Structures

The featurized structures are subjected to a clustering algorithm. We evaluate four clustering algorithms, three popular, representative clustering algorithms, k-means, Gaussian Mixture Model (GMM), and hierarchical clustering, and a variant of the gmx-cluster algorithm in the GROningen MAchine for Chemical Simulations (GROMACS) package [43]. The latter has been shown to be effective in clustering protein structures [44].

We briefly summarize each algorithm next, paying more attention to describing how we optimize their parameters and apply them to the featurized structures.

In k-means, the number of clusters k is a hyper-parameter. The structures that can serve as cluster centroids is another hyper-parameter. We optimize both as follows. For a given value of k, k structures are initially selected uniformly at random over Ω_{gen} to act as the cluster centroids. This induces a particular grouping C of the structures, with each structure assigned to the cluster represented by the structure to which it is closest. To evaluate this particular grouping C, we calculate the within-cluster scatter loss function: $L(C) = \frac{1}{2}\sum_{l=1}^{k}\sum_{i \in C_l}\sum_{j \in C_l, j \neq i} D(x_i, x_j)$, where $D(x_i, x_j)$ measures the Euclidean distance between two points/structures $x_i \neq x_j$ in the same cluster C_l, where $l \in \{1, \ldots, k\}$. One can now vary the structures selected to serve as cluster centroids over iterations and record the selection resulting in the smallest loss. We do so over 10 iterations for a given k, randomly selecting structures as initial centroids in each iteration, recording the optimal selection (and associated grouping) for each iteration.

Note that the above is carried out for a given k as k varies in a permissive range. To find the optimal number of clusters, k, in some considered range, we utilize the popular knee-finding approach [45]. Specifically, after the centroids of clusters are determined (optimally) as above for a given k, the squared distance of each structure in a cluster from the centroid of the cluster can be recorded, and the sum of these squared distances can be obtained over the clusters k [46] . This sum of squared distances is known as the sum of squared errors (SSE) and is shown for a particular structure dataset in Figure 2. In Figure 2, different values of k are plotted against the corresponding SSE values. The knee (also referred to as elbow) in the SSE curve indicates the optimal number of clusters. We are interested in a small value for SSE. Naturally, as one increases k, the SSE approaches 0. It is exactly 0 when $k = |\Omega_{\text{gen}}|$ (every structure is the centroid of its own cluster). The goal is to choose a small value of k that results in a low SSE. The knee or elbow in the curve that tracks SSE as a function of k corresponds to the region where by increasing k, SSE does not change noticeably; this is annotated in Figure 2.

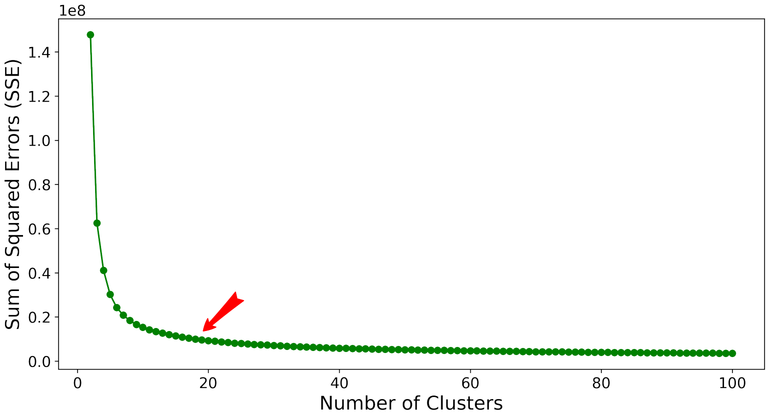

Figure 2. The sum-of-squared errors (SSE) is plotted as a function of the number of clusters k identified via k-means on structures generated via HEA on a target protein. This target is part of our evaluation dataset related in Section 3. Specifically, it is the target protein with known native structure in the PDB entry with identifier (id) 1ail. The red arrow points to the knee/elbow region where by increasing k SSE does not change noticeably; this is the region from where an optimal value of k can be selected.

GMM is a probabilistic model that assumes a mixture of finite number of Gaussian distributions with unknown parameters as the underling process generation of the data. GMM can be thought of as generalizing k-means, as it includes both information from the covariance structure of the data along with the centers of the Gaussian distributions. The main advantage of GMM is the estimation of uncertainty in data membership to clusters; a conditional probability is assigned to each data indicating the probability with which a specific point belongs to any cluster. As expected, sum of all these conditional probabilities for a given point is 1. This uncertainty assignment makes GMM more informative than k-means [47].

However, as in k-means, one needs to specify the number of clusters/components *a priori* in GMM. The optimal value can be determined by minimizing the Bayesian Information Criterion (BIC) [48] metric which considers both covariance type and the number of components. The BIC is a penalty term for the possible likelihood increment when adding more parameters into the model. Specifically, $BIC = \ln(n)k - 2\ln(\hat{L})$, where k is the number of components, $\hat{L}$ is the maximized value of the likelihood function, and n is the number of data points. In Figure 3, we plot the BIC value as the function of the number of components k to demonstrate how one can identify a reasonable value for k at the lowest BIC value.

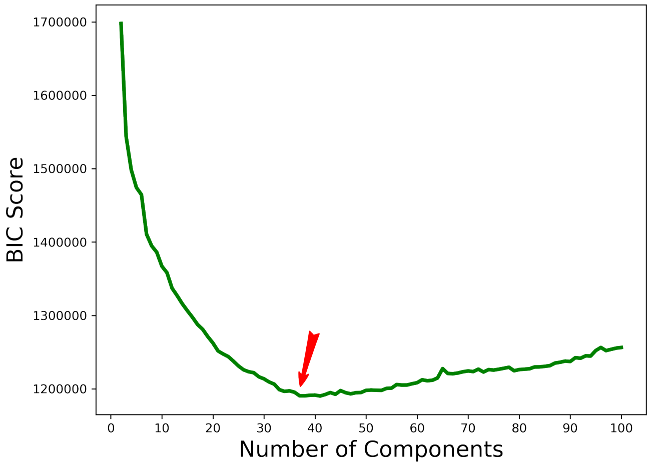

Figure 3. The BIC is plotted as a function of the number of components k. Clustering is carried out via GMM on structures generated via HEA on a target protein (known native structure in the PDB entry with identifier (id) 2h5nd). The red arrow points to the value for k identified at the lowest BIC value.

Unlike k-means and GMM, hierarchical clustering does not require *a priori* specifying the number of clusters. It refers to a family of clustering algorithms that build a sequence of nested clusters by merging or splitting them successively [49]. We make use of the bottom-up (agglomerative) approach for hierarchical clustering; each structure is first in its own clusters, and then clusters are successively merged until the root of the resulting dendrogram is reached, with a unique cluster containing all the data. The linkage criterion specifies the merge strategy. We select single linkage, where the distance between two clusters is defined as the distance between the two closest points across the two clusters [50].

"Cutting" at different locations of the dendrogram results in different partitions of the dataset into clusters. To avoid recomputation of the clusters, we make use of a cached implementation of hierarchical clustering, where cutting the tree at different places does not impose any further computation. We employ the Davies-Bouldin (DB) index [51] to determine where to cut the dendrogram. DB is as popular clustering validation technique in the absence of ground truth labels. It is computed on features inherent to the dataset and gives a measure of the average similarity of each cluster with its most similar cluster. Specifically, the DB index evaluates the intra-cluster similarity and inter-cluster differences to provide a non-negative score. A lower DB index corresponds to a better separation between the clusters. In our application of hierarchical agglomerative clustering with single linkage, we consider the DB index at every height of the tree, and we select the height that results in the smallest DB as the optimal partition (and optimal corresponding number of clusters) of a structure dataset.

Unlike the above clustering algorithms, the gmx-cluster algorithm determines clusters based on a pre-specified distance cutoff. The algorithm first calculates the pairwise distance between all pairs of structures. For each structure x_i, the algorithm then counts the number of other structures (neighbors) that are within the distance cutoff. The structure with the highest number of neighbors is then chosen as the central structure and forms a cluster together with all its neighbors. The formed cluster is then removed from the ensemble of structures and the process is repeated for the remaining structures in the ensemble until the ensemble contains no more structures.

The computation of pairwise distances can potentially be very demanding on large datasets, if one were to use the gmx-cluster implementation that uses lRSMD as the distance metric. Our adaptation of this algorithm transfers all neighbor computations in the USR feature space, using Euclidean distance in the USR feature space as a proxy for lRMSD. These distances, to which we refer as USR scores (and analyze in some detail in Section 4), are normalized between 0 and 1, so that we can set a distance cutoff. We set this cutoff to 0.1; our analysis shows that this is a reasonable value. From now on, we will refer to the adaptation of gmx-cluster as gmx-cluster-usr.

2.4. Stage III: Selecting Structures to Populate the Reduced Ensemble

After clustering the featurized Ω_{gen}, the structures are grouped in clusters. The *selector* now selects a subset of structures from each cluster to populate the reduced ensemble Ω_{red}. The selector makes this decision by considering both the identified clusters and the Rosetta *score*4 energy function of structures. This function evaluates not only Lennard-Jones interactions, but also short- and long-range hydrogen bonding in a tertiary structure.

The selector we propose organizes the structures in a cluster into levels/bins; the structures placed in the same bin have identical score4 energies up to two digits after the decimal sign. One structure is selected at random from each bin and placed in the reduced ensemble Ω_{red}. This process is repeated for each identified cluster.

We note that the selector can control the size of the reduced ensemble by tuning the width of a bin/level. This approach indirectly biases the reduced ensemble by cluster size. Larger clusters with more structures result in more energy levels; therefore, more representative structures are selected from larger clusters to populate the reduced ensemble. Structure diversity retention is another indirect property of this approach as demonstrated experimentally in Section 3.2.

In Section 3, the Ω_{red} ensemble selected as described above is compared against the reduced ensemble identified using truncation selection which does not employ clustering. To populate the reduced ensemble from the truncation-based approach, given a target size M, higher-energy structures are discarded to keep the M lowest-energy structures in an ensemble.

2.5. Datasets for Evaluation

We consider two datasets. The first is a benchmark dataset of 10 proteins of varying lengths (ranging from 53 to 123 amino acids) and folds (α, β, and $\alpha + \beta$) that are widely used for evaluation by structure generation algorithms [12,16,17,52]. The second dataset contains 10 hard, free-modeling target domains from CASP12 and CASP13. To account for stochasticity, the HEA structure generation algorithm is run 5 times for each protein target; the structures generated in each run are aggregated to populate the Ω_{gen} ensemble of 250,000 structures per target. In the clustering algorithms employed here to cluster the featurized Ω_{gen} ensemble, determining the number of clusters takes most of the time. Including the structure generation stage, runtime varies between 7–16 h for a single run on one Intel Xeon E5- 2670 CPU with 2.6 GHz base processing speed and 100 GB of RAM. We note that all our implementations and analyses are carried out in Python. The scikit library is utilized to obtain access to the k-means, GMM, and hierarchical clustering algorithms.

2.6. Experimental Setup

The Ω_{gen} and Ω_{red} ensembles are compared by size, quality, and diversity. First, the evaluation in Section 3 compares the sizes of Ω_{gen} and Ω_{red} on each target protein. The experiment considers each of the four clustering algorithms to determine which algorithm is more effective at size reduction.

To evaluate the proposed selector in our approach against truncation selection, an Ω_{red} ensemble obtained by k-means, GMM, hierarchical clustering, or gmx-cluster-usr is compared to the Ω_{red} ensemble obtained via truncation selection. We choose the maximum size over Ω_{red} identified by k-means, GMM, hierarchical clustering, and gmx-cluster-usr to set the target size M for truncation selection. As the results in Section 3 demonstrate, k-means and GMM result in larger, reduced ensembles compared to those obtained with the hierarchical clustering or gmx-cluster-usr; therefore, the size of the reduced ensemble in the truncation based-approach matches the size of the reduced ensemble obtained via k-means or GMM.

Second, the Ω_{gen} and Ω_{red} ensembles on each target protein are compared in terms of the distances of the structures in them from a known native structure. As is common practice in *de-novo* PSP, we utilize the popular least root-mean-squared-deviation (lRMSD) to measure these distances [53]. We note that lRMSD stands for least RMSD, which refers to a two-step process, first removing the rigid-body translation and rotation in three-dimensional space (rigid-body motions) between two structures under comparison and then calculating the weighted/average Euclidean distances of corresponding atoms in the structures. We report the lRMSD measured over the main carbon atoms (CA atoms). The evaluation in Section 3 compares the minimum, average, and standard deviation of lRMSDs of structures from the known native structure in the Ω_{gen} and Ω_{red} ensembles on each target protein.

Third, the Ω_{gen} and Ω_{red} ensembles on each target protein are compared in terms of their energetic profiles. While the native structure is a reliable reference structure for an lRMSD-based comparison between two ensembles, the same is not true for energy. The native structure, as obtained from the PDB, has higher energy than many computed structures, as it has not been refined for a particular energy function. Refining it until a local optimum of the energy function under consideration has been obtained will change the native structure, often in significant ways. Therefore, the evaluation in Section 3 compares the energy distributions of the Ω_{gen} and Ω_{red} ensembles directly, and relates observations regarding the minimum and average energies of the ensembles, as well as the maximum energy difference between two structures in an ensemble.

Finally, the lRMSD and energy distributions are related jointly. Each structure is visualized based on its lRMSD from the known native structure serving as one coordinate and its Rosetta score4 energy serving as the other coordinate. The so-called energy landscapes associated with the Ω_{gen} and Ω_{red} ensembles of a target protein are visualized, and superimpositions are utilized to identify possible regions of the landscape that the proposed approach may have discarded.

3. Results

3.1. Comparing Ensemble Sizes Pre- and Post Reduction

In Table 1, Ω_{gen} and Ω_{red} are first compared in terms of size over the benchmark dataset. The reduction percentage $(1 - \frac{|\Omega_{\text{red}}|}{|\Omega_{\text{gen}}|}) \cdot 100\%$ is also reported for each target. The reductions obtained by k-means range from 54% to 71%. The GMM reductions vary from 53% to 71%. Gmx-cluster-usr and hierarchical clustering result in more dramatic reductions of more than 72% and 77% on all targets respectively, and over 80% on 5/10 and 9/10 of the targets, respectively. Similar results are obtained over the CASP dataset, as shown in Table 2. Reductions of 59% and higher are obtained via k-means. Reductions obtained via GMM are comparable to those obtained via k-means. Reductions of 72% and higher are achieved via gmx-cluster-usr. Reductions of around 80% and higher are obtained via hierarchical clustering.

Table 1. Ω_{gen} and Ω_{red} are compared in terms of size over the benchmark dataset. The PDB ids, lengths, and folds of the dataset are shown in Columns 1–3. Column 4 shows the size of Ω_{gen}. The size of Ω_{red} and the reduction of Ω_{red} over Ω_{gen} are shown in Columns 5–12 for all clustering algorithms.

				K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**	
PDB Id	**Length**	**Fold**	$\lvert\Omega_{\text{gen}}\rvert$	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**
1ail	70	α	250 K	94,867	62.05	99,707	60.12	32,432	87.03	48,450	80.62
1bq9	53	β	250 K	79,181	68.33	77,873	68.85	24,705	90.12	39,716	84.11
1c8ca	64	β	250 K	87,209	65.12	88,437	64.63	29,795	88.08	46,817	81.27
1cc5	83	α	250 K	97,878	60.85	101,589	59.36	36,630	85.35	55,673	77.73
1dtja	76	$\alpha + \beta$	250 K	75, 421	69.83	79,134	68.35	29,506	88.2	41,456	83.42
1hhp	99	β	250 K	71,926	71.23	71,390	71.44	27,208	89.12	42,226	83.11
1tig	88	$\alpha + \beta$	250 K	94,656	62.14	97,010	61.2	40,145	83.94	57,033	77.19
2ezk	93	α	250 K	114,244	54.3	115,929	53.63	49,509	80.2	62,439	75.02
2h5nd	123	α	250 K	110,196	55.92	111,353	55.46	55,153	77.94	67,671	72.93
3gwl	106	α	250 K	101,827	59.27	105,214	57.91	46,480	81.41	63,116	74.75

Table 2. Ω_{gen} and Ω_{red} are compared in terms of size over the CASP dataset. The CASP ids and lengths are shown in Columns 1–2. Column 3 shows the size of Ω_{gen}. The size of Ω_{red} and the reduction of Ω_{red} over Ω_{gen} are shown in Columns 4–11 for all clustering algorithms.

			K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**	
CASP Id	**Length**	$\lvert\Omega_{\text{gen}}\rvert$	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**	$\lvert\Omega_{\text{red}}\rvert$	**Red. (%)**
T0859-D1	129	250 K	91,236	63.51	94,014	62.39	32,060	87.18	50,903	79.64
T0886-D1	69	250 K	72,351	71.06	88,986	64.41	27,328	89.07	42,397	83.04
T0892-D2	110	250 K	89,943	64.02	92,200	63.12	39,669	84.13	55,482	77.81
T0897-D1	138	250 K	98,262	60.7	101,119	59.55	50,352	79.86	68,703	72.52
T0898-D2	55	250 K	67,046	73.18	67,283	73.09	21,332	91.47	35,053	85.98
T0953s1-D1	67	250 K	51,078	79.57	50,509	79.8	16, 417	93.43	29,690	88.12
T0953s2-D3	93	250 K	73,191	70.72	74,974	70.01	22,143	91.14	38,372	84.65
T0957s1-D1	108	250 K	92,028	63.19	93,872	62.45	38,665	84.53	54,951	78.02
T0960-D2	84	250 K	53,388	78.64	52,136	79.15	22,171	91.13	32,548	86.98
T1008-D1	77	250 K	101,433	59.43	105,360	57.86	51,809	79.28	68,428	72.63

3.2. Comparing Distributions of lRMSDs from the Native Structure Pre- and Post Reduction

Table 3 compares the minimum, average, and standard deviation of lRMSDs of structures in the Ω_{red} and Ω_{gen} ensembles to the known native structure on each target in the benchmark dataset. The top panel of the table compares the minimum lRMSDs of the ensembles including the ensemble generated by truncation-based selection; the middle panel compares the average lRMSDs of the ensembles, and the bottom one compares the standard deviation of lRMSDs of structures in each ensemble to the known native structure per target protein. The minimum, average, and standard deviations over the generated ensembles are provided as reference in Column 2 (top, middle, and bottom panels, respectively). The difference of the (lRMSD) minimum, average, or standard deviation in Ω_{red} over the corresponding quantity in Ω_{gen} is reported in each setting.

Table 3. Comparison of minimum, average, and standard deviation of lRMSDs (to the known native structure) of structures in the Ω_{gen} and Ω_{red} ensembles of each target in the benchmark dataset. Comparison of minimum lRMSDs includes the ensemble reduced via truncation selection. Differences between the minimum, average, and standard deviation obtained over Ω_{red} from those obtained over Ω_{gen} are also related.

Minimum lRMSD (Å)											
		K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**		**Truncation**	
PDB Id	Ω_{gen}	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**
1ail	3.64	3.64	0	3.64	0	3.64	0	3.64	0	4.37	0.73
1bq9	5.42	5.42	0	5.42	0	5.47	0.05	5.47	0.05	7.31	1.89
1c8ca	4.43	4.43	0	4.43	0	4.43	0	4.43	0	7.86	3.43
1cc5	5.4	5.4	0	5.4	0	6.52	1.12	5.4	0	7.85	2.45
1dtja	4.19	4.19	0	4.19	0	4.19	0	4.19	0	9.31	5.12
1hhp	11	11	0	11	0	11.29	0.29	11.02	0.02	12.88	1.88
1tig	5.34	5.34	0	5.34	0	5.45	0.11	5.45	0.11	6.59	1.25
2ezk	3.41	3.41	0	3.41	0	3.41	0	3.41	0	5.09	1.68
2h5nd	10.32	10.32	0	10.32	0	10.32	0	10.32	0	11.9	1.58
3gwl	4.85	4.85	0	4.85	0	4.85	0	4.85	0	7.81	2.96

Average lRMSD (Å)									
		K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**	
PDB Id	Ω_{gen}	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**
1ail	10.42	10.89	0.47	10.85	0.43	10.19	0.23	11.17	0.75
1bq9	9.76	10.18	0.42	10.17	0.41	9.74	0.02	10.39	0.63
1c8ca	12.29	12.75	0.46	12.72	0.43	12.25	0.04	12.96	0.67
1cc5	12.53	12.94	0.41	12.89	0.36	12.5	0.03	13.12	0.59
1dtja	11.87	12.35	0.48	12.29	0.42	11.95	0.08	12.5	0.63
1hhp	15.56	16.06	0.5	16.03	0.47	15.85	0.29	16.31	0.75
1tig	12.85	13.36	0.51	13.31	0.46	12.81	0.04	13.81	0.96
2ezk	10.17	10.77	0.6	10.72	0.55	10.78	0.61	11.21	1.04
2h5nd	15.79	16.24	0.45	16.2	0.41	16.16	0.37	16.51	0.72
3gwl	12.44	13.02	0.58	12.94	0.5	12.68	0.24	13.34	0.9

Standard Deviation lRMSD (Å)									
		K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**	
PDB Id	Ω_{gen}	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**
1ail	3.11	3.2	0.09	3.17	0.06	3.11	0	3.49	0.38
1bq9	1.78	1.76	0.02	1.89	0.11	1.89	0.11	2.11	0.33
1c8ca	2.18	2.23	0.05	2.22	0.04	2.15	0.03	2.45	0.27
1cc5	2.31	2.34	0.03	2.34	0.03	2.35	0.04	2.54	0.23
1dtja	2.08	2.31	0.23	2.27	0.19	2.15	0.07	2.44	0.36
1hhp	1.82	1.94	0.12	1.93	0.11	1.9	0.08	2.08	0.26
1tig	3.22	3.34	0.12	3.34	0.12	3.21	0.01	3.72	0.5
2ezk	2.41	2.67	0.26	2.66	0.25	2.77	0.36	2.94	0.53
2h5nd	2.03	2.22	0.19	2.21	0.18	2.3	0.27	2.46	0.43
3gwl	2.9	3.02	0.12	3.01	0.11	2.99	0.09	3.22	0.32

Focusing on the *Diff.* columns listing differences in minimum lRMSDs, it is clear that truncation selection performs the worst in this regard; differences in minimum lRMSD range from 0.73 Å to 5.12 Å (see Column 12 of the top panel in Table 3). This means in the worst case, the best structure kept by truncation selection is 5.12 Å further away from the native structure than the best structure in the original ensemble. Truncation-based selection cannot maintain the quality of original ensemble.

In contrast, in the case of GMM and k-means, differences in minimum lRMSD (see Columns 4 and 6) are all 0 Å. The differences in minimum lRMSD for gmx-cluster-usr range from 0 Å to 0.11 Å (see Column 10); for hierarchical clustering, the range is from 0 Å to 1.12 Å (see Column 8). This means that the structures closest to the native structure are always retained in the ensembles reduced by k-means and GMM. Not surprisingly, the slight increase in the differences when using gmx-cluster-usr and hierarchical clustering is the consequence of more drastic reduction in size of the reduced ensemble when using these two clustering algorithms over k-means or GMM.

The comparison shown in the middle panel of Table 3 indicates very little difference between the generated and reduced ensembles in terms of average lRMSDs. Column 4 shows the differences in average lRMSDs for k-means, which range from 0.41 Å to 0.60 Å. Column 6 shows an overall similar range for GMM (0.36 Å to 0.50 Å). Average lRMSD differences for hierarchical clustering range from 0.02 Å to 0.61 Å, as shown in Column 8. Column 10 shows that the differences in average lRMSD for gmx-cluster-usr range from 0.59 Å to 1.04 Å.

The comparison of differences on lRMSDs standard deviation for k-means is shown in Column 4 on the bottom panel and vary from 0.02 Å to 0.26 Å. These values are slightly different for GMM, ranging from 0.03 Å to 0.25 Å(see Column 6). As in the minimum lRMSD comparison, the differences obtained by gmx-cluster-usr and hierarchical clustering are slightly larger. Differences in standard deviation range from 0.23 Å to 0.53 Å for gmx-cluster-usr (shown in Column 10) and from 0 Å to 0.36 Å (with less than 0.1 Å on 7/10 targets; shown in Column 8) for hierarchical clustering.

Similar observations can be extracted from Table 4, which shows the performance over the CASP dataset. Table 4 confirms again that truncation selection loses the quality of the original ensemble in the reduced one. The quality of the reduced ensemble is preserved by all clustering algorithms, and the best results belong to k-means and GMM. All four clustering algorithms produce ensembles that have small differences in average lRMSDs and perform comparably in terms of standard deviation.

Greater detail can be inferred from Figure 4, which shows results over a selected target protein (with native structure under PDB id 1ail). Figure 4 shows the actual distribution of structure lRMSDs from the native structure for the Ω_{gen} ensemble along with the ensembles Ω_{red} reduced via k-means, GMM, gmx-cluster-usr, and hierarchical clustering. Figure 4 shows that structures with similar relative frequencies of lRMSDs as in Ω_{gen} are included in the reduced ensembles identified by each clustering algorithm.

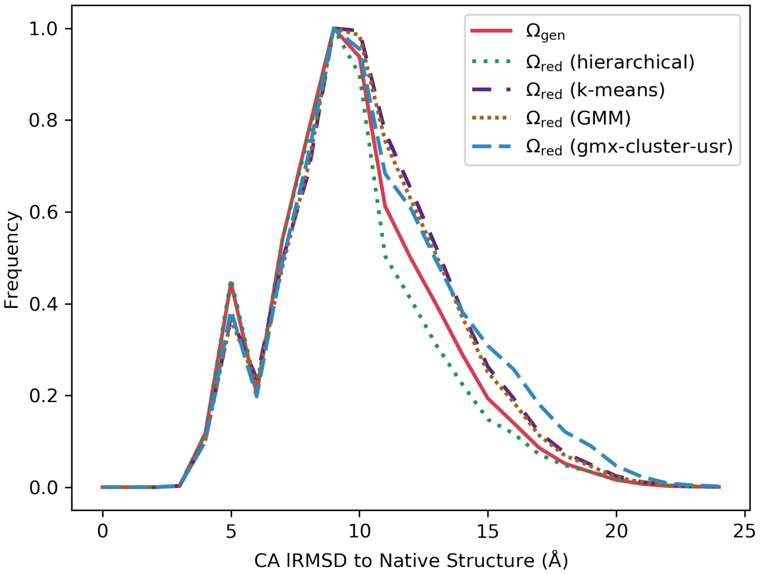

Figure 4. The distribution of structure lRMSDs from the native structure is shown for the Ω_{gen} ensemble (in red) and the reduced Ω_{red} ensembles obtained via k-means (purple), GMM (brown), hierarchical clustering (green), and gmx-cluster-usr (in blue). Results are shown for a representative target protein with native structure under PDB id 1ail.

Table 4. Comparison of minimum, average, and standard deviation of distribution of lRMSDs (to the known native structure) of structures in the Ω_{gen} and Ω_{red} ensembles of each target in the CASP dataset. Comparison of minimum lRMSDs includes the ensemble reduced via truncation selection. Differences between the minimum, average, and standard deviation obtained over Ω_{red} from those obtained over Ω_{gen} are also related.

Minimum lRMSD (Å)											
		K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**		**Truncation**	
CASP Id	Ω_{gen}	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**
T0859-D1	11.37	11.37	0	11.37	0	11.96	0.59	11.96	0.59	13.12	1.75
T0886-D1	7.96	7.96	0	7.96	0	8.73	0.77	8.73	0.77	11.24	3.28
T0892-D2	7.71	7.71	0	7.71	0	8.28	0.57	7.71	0	9.11	1.4
T0897-D1	10.18	10.18	0	10.18	0	10.64	0.46	10.64	0.46	11.62	1.44
T0898-D2	7.51	7.51	0	7.51	0	7.51	0	7.51	0	8.68	1.17
T0953s1-D1	6.14	6.14	0	6.14	0	6.29	0.15	6.29	0.15	8.18	2.04
T0953s2-D3	7.13	7.13	0	7.13	0	7.24	0.11	7.24	0.11	8.17	1.04
T0957s1-D1	7.65	7.65	0	7.65	0	7.76	0.11	7.76	0.11	9.39	1.74
T0960-D2	7.26	7.26	0	7.26	0	7.26	0	7.26	0	8.12	0.86
T1008-D1	3.85	3.85	0	3.85	0	3.85	0	3.85	0	5.67	1.82

Minimum lRMSD (Å)									
		K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**	
CASP Id	Ω_{gen}	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**
T0859-D1	17.47	17.64	0.17	17.63	0.16	17.49	0.02	17.78	0.31
T0886-D1	13.16	13.66	0.5	13.67	0.51	13.31	0.15	13.82	0.66
T0892-D2	14.81	15.49	0.68	15.43	0.62	15.02	0.21	15.71	0.9
T0897-D1	17.3	17.84	0.54	17.81	0.51	17.54	0.24	18.04	0.74
T0898-D2	11.56	11.72	0.16	11.71	0.15	11.63	0.07	11.86	0.3
T0953s1-D1	11.98	11.74	0.24	11.73	0.25	11.81	0.17	11.66	0.32
T0953s2-D3	13.28	13.89	0.61	13.88	0.6	13.48	0.2	14.06	0.78
T0957s1-D1	14.96	15.49	0.53	15.44	0.48	15.13	0.17	15.74	0.78
T0960-D2	12.63	13.07	0.44	13.07	0.44	12.93	0.3	13.27	0.64
T1008-D1	11.77	12.36	0.59	12.46	0.69	11.9	0.13	12.67	0.9

Minimum lRMSD (Å)									
		K-Means		**GMM**		**Hierarchical**		**Gmx-Cluster-Usr**	
CASP Id	Ω_{gen}	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**	Ω_{red}	**Diff.**
T0859-D1	1.83	1.84	0.01	1.9	0.07	1.91	0.08	1.99	0.16
T0886-D1	1.67	1.82	0.15	1.79	0.12	1.69	0.02	1.98	0.31
T0892-D2	3.02	2.98	0.04	2.97	0.05	3.04	0.02	3.25	0.23
T0897-D1	2.75	2.74	0.01	2.73	0.02	2.76	0.01	2.92	0.17
T0898-D2	1.03	1.17	0.14	1.16	0.13	1.14	0.11	1.26	0.23
T0953s1-D1	1.51	1.51	0	1.51	0	1.5	0.01	1.49	0.02
T0953s2-D3	1.9	1.84	0.06	1.83	0.07	1.86	0.04	2.06	0.16
T0957s1-D1	3.04	3.04	0	3.03	0.01	3.04	0	3.23	0.19
T0960-D2	1.85	1.94	0.09	1.95	0.1	1.92	0.07	2.05	0.2
T1008-D1	3.7	3.72	0.02	3.69	0.01	3.74	0.04	3.94	0.24

3.3. Comparing Distributions of Energies Pre- and Post Reduction

Comparison of the minimum energy in an ensemble prior to and after reduction reveals some expected results. Since the proposed approach selects structures from each energy level, differences in the minimum energy in the Ω_{red} and Ω_{gen} ensembles of any target protein are negligible (less than 10^{-2}). Since truncation selection retains the lowest-energy structure, as well, such differences are 0. Similar observations hold when comparing the energy diameters (the difference between the highest and lowest energy in an ensemble) of the Ω_{red} and Ω_{gen} ensembles on each target protein for the proposed approach. The differences are negligible (less than 10^{-2}), as the proposed approach selects structures from all energy levels. The same is not true of truncation selection. The original ensemble contains many higher-energy structures, which truncation selection discards. As expected, for the same reasons, comparing the average energy yields small differences for the proposed approach while truncation selection results in large differences.

Figure 5 shows the distribution of structure energies for the Ω_{gen} ensemble along with the Ω_{red} ensembles reduced via k-means, GMM, gmx-cluster-usr, and hierarchical clustering for a selected target protein (with native structure under PDB id 1ail). Figure 5 shows that the relative frequencies of energies of the structures in the reduced ensembles are close to that of the structures in Ω_{gen}.

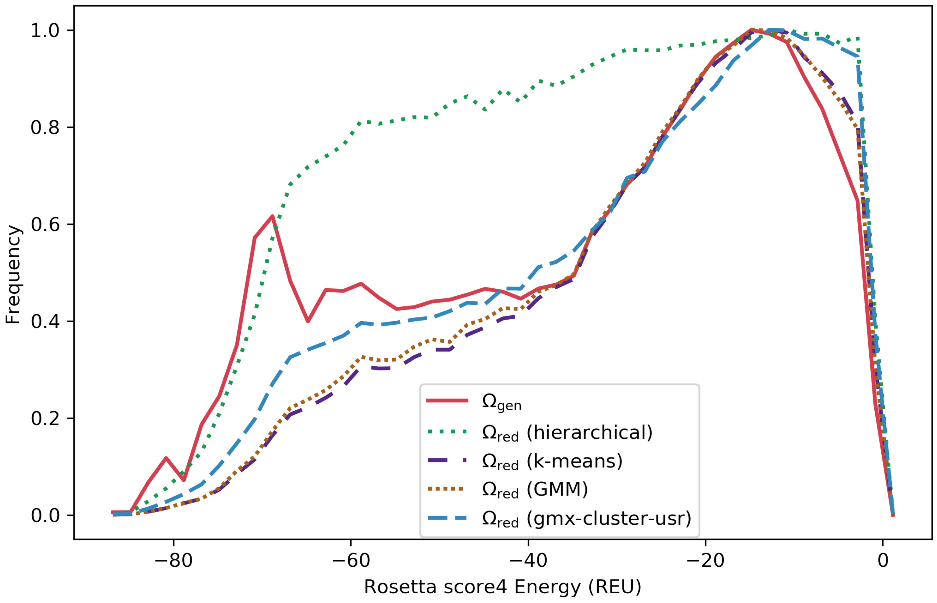

Figure 5. The distribution of Rosetta score4 energies is shown for the Ω_{gen} ensemble (in red) and the reduced Ω_{red} ensembles obtained via k-means (purple), GMM (brown), hierarchical (green), and gmx-cluster-usr clustering (in blue). Results are shown for the target protein with native structure under PDB id 1ail.

3.4. Visually Comparing Distributions of lRMSDs and Energies Pre- and Post Reduction

We now compare the Ω_{gen} and Ω_{red} ensembles on each target protein in terms of Rosetta score4 energies and lRMSDs to the native structure. While the Supplementary Material shows these landscape figures for each target, here we show one representative landscape on each dataset (benchmark and CASP) that illustrates the behavior of each of the clsutering algorithms. Structures in Ω_{gen} are drawn in purple, while those in the Ω_{red} ensemble are drawn in green. Figure 6 does so for the benchmark dataset, and Figure 7 does so for the CASP dataset.

Figures 6 and 7 (and those shown for each target protein in the Supplementary Material) show that the reduced ensemble Ω_{red} includes structures from all the regions in the structure space populated by the original ensemble Ω_{gen}. All the purple dots being occluded by the superimposition in the k-means and GMM case visually makes the case that these two clustering algorithms perform better than gmx-cluster-usr and hierarchical clustering. This is not surprising, as k-means and GMM preserve more of the original ensemble.

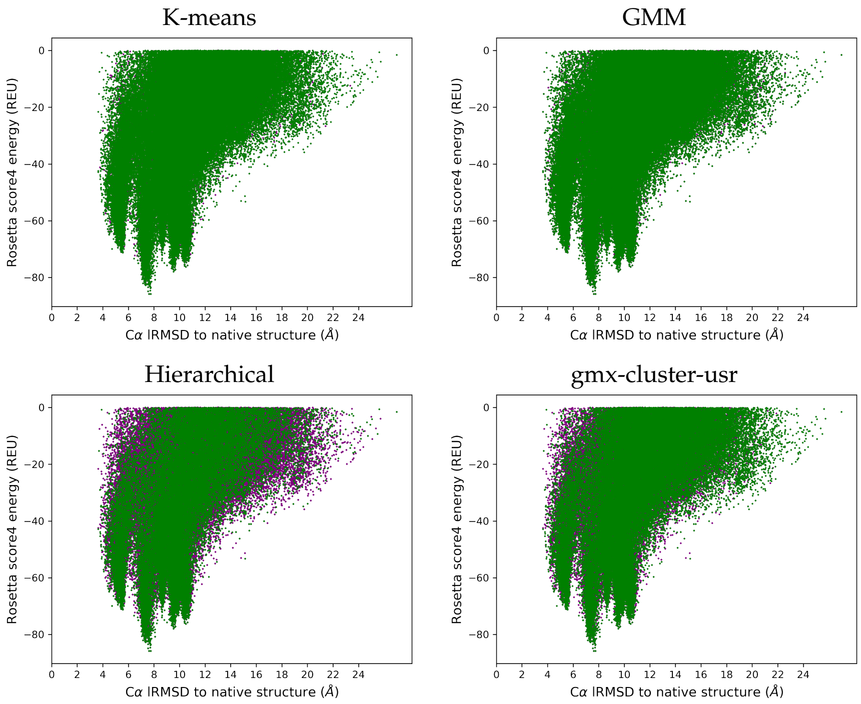

Figure 6. Benchmark Dataset: A representative target (with known native structure under PDB id 1ail) is selected. Structures in the Ω_{gen} ensemble are plotted in purple in terms of their lRMSD (Å) from the native structure (X-axis) versus their Rosetta score4 energies (Y-axis) measured in Rosetta Energy Units (REUs). Structures in the Ω_{red} ensemble are superimposed in green.

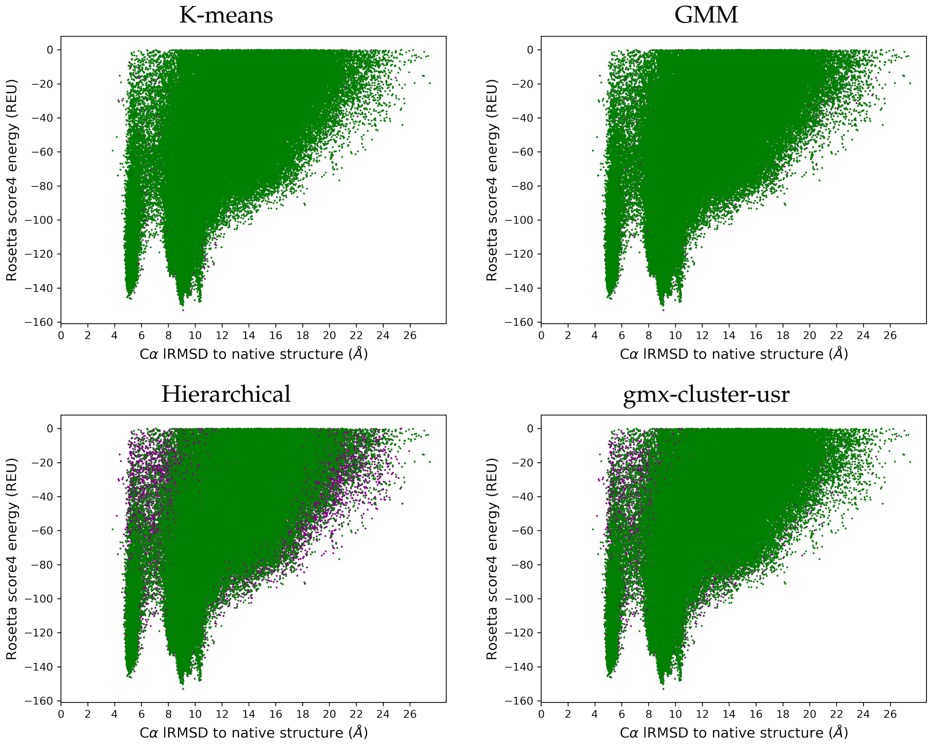

Figure 7. CASP Dataset: A representative protein (T1008-D1) is selected. Structures in the Ω_{gen} ensemble are plotted in purple in terms of their lRMSD (Å) from the native structure (X-axis)versus their Rosetta score4 energies (Y-axis) measured in Rosetta Energy Units (REUs). Structures in the Ω_{red} ensemble are superimposed in green.

4. Discussion

In summary, the presented results make the case that all four clustering algorithms are able to drastically decrease the structure ensemble size while preserving the quality and diversity of the original ensemble. GMM and k-means behave equally well in this regard, while gmx-cluster-usr and hierarchical clustering reduces the size of the ensembles more significantly and in response is also more prone to sacrificing quality.

Experience in molecular structure modeling informs that the choice of representation is often key to the success of a method. Here we provide further analysis into what the USR features are capturing. We do via a simple correlation analysis, where we compare the distribution of the lRMSDs versus USR scores of computed tertiary structures to the native structure. USR score is calculated as the Euclidean distance in the 12-dimensional USR feature space for two structures. The Supplementary Material (Table S5) lists the Pearson's correlation coefficient between the two distributions for each protein target (in both the benchmark and CASP datasets).

Figure 8 plots the distributions against each-other for two targets that are representatives of the Pearson's correlation coefficients obtained over targets in the benchmark and CASP datasets. Specifically, Figure 8a shows a correlation of 0.80 that is representative of what is observed over the benchmark dataset; Figure 8b shows a correlation of 0.74 that is representative of what is observed over the CASP dataset.

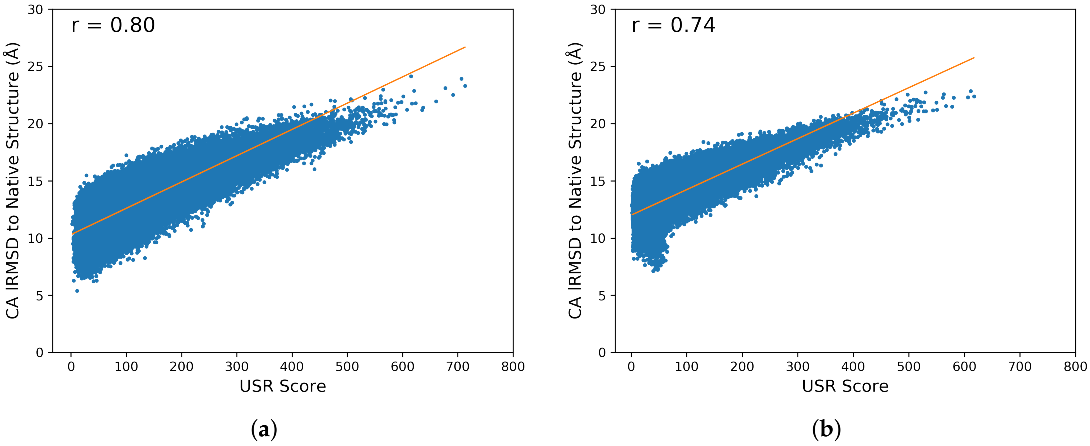

Figure 8. Correlation between USR scores and lRMSDs to the native structure of all tertiary structures computed on a target protein in the (**a**) benchmark dataset (with native structure under PDB id 1cc5) and (**b**) CASP dataset (with native structure under CASP id T0953s2-D3.)

The median correlation over the benchmark dataset is 0.80, and the median correlation over the CASP dataset is 0.755. The correlations (representing what is observed over each of the datasets, with few outliers) show that the USR score is informative and a good proxy for lRMSD; we recall that the USR representation is also invariant to rigid-body motions, unlike Cartesian coordinate-based representations. Altogether, these results inform that the choice of the USR-based representation of tertiary structures is advantageous, allowing clustering algorithms to capture important structural differences that are then retained in the reduced ensemble by the selector.

5. Conclusions

The findings presented in this paper suggest that it is possible to significantly reduce the number of generated decoys without sacrificing quality and diversity. A three-stage approach relying on featurization, clustering, and selection is shown effective at doing so independent of the particular clustering algorithm employed. Various clustering algorithms are evaluated in the proposed approach.

The presented work opens up many venues of further research. An interesting application of this work can be in preparing data for model assessment methods, particularly in cases where datasets are expected to be highly imbalanced and so affect the performance of these methods. Other directions concern evaluating more clustering algorithms, including, for instance, clique finding algorithms. As often the case in molecular structure modeling and analysis literature, an important direction remains the choice of representation. Other future work can therefore investigate the utility and effectiveness of different features to represent structures, as well as consider advances in subspace clustering to address the high-dimensionality of molecular structure spaces.

Supplementary Materials: Supplementary Materials are available online.

Author Contributions: A.B.Z. conceptualized and implemented the methodologies described here, carried out the evaluation, and drafted the manuscript. P.K. assisted with implementation and evaluation of the methodologies and editing of the manuscript. C.D. and A.S. guided the research, conceptualization, evaluation, and edited and finalized the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by NSF Grant No. 1763233 and a Jeffress Memorial Trust Award.

Acknowledgments: Computations were run on ARGO, a research computing cluster provided by the Office of Research Computing at George Mason University. We thank members of the Shehu laboratory for useful feedback on this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BIC	Bayesian Information Criterion
CASP	Critical Assessment of protein Structure Prediction
DB	Davies-Bouldin
GMM	Gaussian Mixture Model
HEA	Hybrid Evolutionary Algorithm
lRMSD	least Root-Mean-Squared-Deviation
PDB	Protein Data Bank
PSP	Protein Structure Prediction
SSE	Sum of Squared Errors
USR	Ultrafast Shape Recognition
GROMACS	GROningen MAchine for Chemical Simulations

References

1. Boehr, D.D.; Wright, P.E. How do proteins interact? *Science* **2008**, *320*, 1429–1430. [CrossRef]
2. Maximova, T.; Moffatt, R.; Ma, B.; Nussinov, R.; Shehu, A. Principles and Overview of Sampling Methods for Modeling Macromolecular Structure and Dynamics. *PLoS Comput. Biol.* **2016**, *12*, e1004619. [CrossRef]
3. Blaby-Haas, C.E.; de Crécy-Lagard, V. Mining high-throughput experimental data to link gene and function. *Trends Biotechnol.* **2013**, *29*, 174–182. [CrossRef] [PubMed]
4. Berman, H.M.; Henrick, K.; Nakamura, H. Announcing the worldwide Protein Data Bank. *Nat. Struct. Biol.* **2003**, *10*, 980–980. [CrossRef]
5. Lee, J.; Freddolino, P.; Zhang, Y. Ab initio protein structure prediction. In *From Protein Structure to Function with Bioinformatics*, 2nd ed.; Rigden, D.J., Ed.; Springer: London, UK, 2017; Chapter 1, pp. 3–35.
6. Levitt, M. Nature of the protein universe. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 11079–11084. [CrossRef]
7. Das, R. Four small puzzles that Rosetta doesn't solve. *PLoS ONE* **2011**, *6*, e20044. [CrossRef] [PubMed]
8. Molloy, K.; Saleh, S.; Shehu, A. Probabilistic Search and Energy Guidance for Biased Decoy Sampling in Ab-initio Protein Structure Prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2013**, *10*, 1162–1175. [CrossRef] [PubMed]
9. Akhter, N.; Qiao, W.; Shehu, A. An Energy Landscape Treatment of Decoy Selection in Template-free Protein Structure Prediction. *Computation* **2018**, *6*, 39. [CrossRef]
10. Leaver-Fay, A.; Tyka, M.; Lewis, S.M.; Lange, O.F.; Thompson, J.; Jacak, R.; Kaufman, K.; Renfrew, P.D.; Smith, C.A.; Sheffler, W.; et al. ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. *Methods Enzym.* **2011**, *487*, 545–574.

11. Xu, D.; Zhang, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins Struct. Funct. Bioinform.* **2012**, *80*, 1715–1735. [CrossRef]
12. Zaman, A.; Shehu, A. Balancing multiple objectives in conformation sampling to control decoy diversity in template-free protein structure prediction. *BMC Bioinform.* **2019**, *20*, 211. [CrossRef] [PubMed]
13. Zaman, A.; De Jong, K.A.; Shehu, A. Using Subpopulation EAs to Map Molecular Structure Landscapes. In Proceedings of the Genetic and Evolutionary Computation (GECCO), Prague, Czech Republic, 13–17 July 2019; ACM: New York, NY, USA, 2019; pp. 1–8.
14. Olson, B.; Shehu, A. Multi-Objective Optimization Techniques for Conformational Sampling in Template-Free Protein Structure Prediction. In Proceedings of the Bioinform and Comp Biol (BICoB), Las Vegas, NV, USA, 24–26 March 2014; pp. 143–148.
15. Olson, B.; De Jong, K.A.; Shehu, A. Off-Lattice Protein Structure Prediction with Homologous Crossover. In Proceedings of the Genetic and Evolutionary Computation (GECCO), Amsterdam, The Netherlands, 6–10 July 2013; ACM: New York, NY, USA, 2013; pp. 287–294.
16. Olson, B.; Shehu, A. Multi-Objective Stochastic Search for Sampling Local Minima in the Protein Energy Surface. In Proceedings of the Bioinf and Comp Biol (BCB), Washington, DC, USA, 22–25 September 2013; pp. 430–439.
17. Zhang, G.; Ma, L.; Wang, X.; Zhou, X. Secondary Structure and Contact Guided Differential Evolution for Protein Structure Prediction. *IEEE Trans. Comput. Biol. Bioinform.* **2018**, *1*. [CrossRef] [PubMed]
18. Zhang, C.; Mortuza, S.M.; He, B.; Wang, Y.; Zhang, Y. Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12. *Proteins Struct. Funct. Bioinform.* **2018**, *86*, 136–151. [CrossRef] [PubMed]
19. Zaman, A.; Parthasarathy, P.V.; Shehu, A. Using Sequence-Predicted Contacts to Guide Template-free Protein Structure Prediction. In Proceedings of the Bioinf and Comp Biol (BCB), Niagara Falls, NY, USA, 7–10 September 2019; ACM: Niagara Falls, NY, USA, 2019; pp. 154–160.
20. Hou, J.; Wu, T.; Cao, R.; Cheng, J. Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. *Proteins* **2019**, *87*, 1165–1178. [CrossRef] [PubMed]
21. Schaarschmidt, J.; Monastyrskyy, B.; Kryshtafovych, A.; Bonvin, A. Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age. *Proteins* **2018**, *86*, 51–66. [CrossRef] [PubMed]
22. Cheung, N.J.; Yu, W. De novo protein structure prediction using ultra-fast molecular dynamics simulation. *PLoS ONE* **2018**, *13*, e0205819. [CrossRef]
23. Gao, M.; Zhou, H.; Skolnick, J. DESTINI: A deep-learning approach to contact-driven protein structure prediction. *Sci Rep.* **2019**, *9*, 3514. [CrossRef]
24. Senior, A.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A.; Bridgland, A.; et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706–710. [CrossRef]
25. Kryshtafovych, A.; Monastyrskyy, B.; Fidelis, K.; Schwede, T.; Tramontano, A. Assessment of model accuracy estimations in CASP12. *Proteins Struct. Funct. Bioinfom.* **2017**, *86*, 345–360. [CrossRef]
26. Shehu, A. A Review of Evolutionary Algorithms for Computing Functional Conformations of Protein Molecules. In *Computer-Aided Drug Discovery;* Zhang, W., Ed.; Methods in Pharmacology and Toxicology; Springer: Berlin/Heidelberg, Germany, 2015.
27. Zhang, Y.; Skolnick, J. SPICKER: A clustering approach to identify near-native protein folds. *J. Comput. Chem.* **2004**, *25*, 865–871. [CrossRef]
28. Akhter, N.; Chennupati, G.; Djidjev, H.; Shehu, A. Decoy Selection for Protein Structure Prediction Via Extreme Gradient Boosting and Ranking. *BMC Bioinform.* **2020**, in press.
29. Kabir, L.K.; Hassan, L.; Rajabi, Z.; Shehu, A. Graph-based Community Detection for Decoy Selection in Template-free Protein Structure Prediction. *Molecules* **2019**, *24*, 854. [CrossRef] [PubMed]
30. Akhter, N.; Shehu, A. From Extraction of Local Structures of Protein Energy Landscapes to Improved Decoy Selection in Template-free Protein Structure Prediction. *Molecules* **2018**, *23*, 216. [CrossRef] [PubMed]
31. Shortle, D.; Simons, K.T.; Baker, D. Clustering of low-energy conformations near the native structures of small proteins. *Proc. Natl. Acad. Sci. USA* **1998**, *95*, 11158–11162. [CrossRef]
32. Li, S.; Ng, Y. Calibur: A tool for clustering large numbers of protein decoys. *BMC Bioinform.* **2010**, *11*, 25. [CrossRef]

33. Li, S.C.; Bu, D.; Li, M. Clustering 100,000 Protein Structure Decoys in Minutes. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2012**, *9*, 765–773. [CrossRef]
34. Zaman, A.; Kamranfar, P.; Domeniconi, C.; Shehu, A. Decoy Ensemble Reduction in Template-Free Protein Structure Prediction. In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Niagara Falls, NY, USA, 7–10 September 2019; ACM: Niagara Falls, NY, USA, 2019; pp. 562–567.
35. Ballester, P.J.; Richards, W.G. Ultrafast shape recognition to search compound databases for similar molecular shapes. *J. Comput. Chem.* **2007**, *28*, 1711–1723. [CrossRef]
36. Shehu, A. An Ab-initio tree-based exploration to enhance sampling of low-energy protein conformations. In *Robotics: Science and Systems V*; Trinkle, J., Matsuoka, Y., Castellanos, J.A., Eds.; The MIT Press: Cambridge, MA, USA, 2009; pp. 241–248.
37. Olson, B.; Shehu, A. Guiding the Search for Native-like Protein Conformations with an Ab-initio Tree-based Exploration. *Int. J. Robot. Res.* **2010**, *29*, 1106–1127.
38. Molloy, K.; Shehu, A. Interleaving Global and Local Search for Protein Motion Computation. In *LNCS: Bioinformatics Research and Applications*; Harrison, R., Li, Y., Mandoiu, I., Eds.; Springer International Publishing: Norfolk, VA, USA, 2015; Volume 9096, pp. 175–186.
39. Molloy, K.; Shehu, A. A General, Adaptive, Roadmap-based Algorithm for Protein Motion Computation. *IEEE Trans. NanoBiosci.* **2016**, *2*, 158–165. [CrossRef]
40. Mani, P.; Vazquez, M.; Metcalf-Burton, J.R.; Domeniconi, C.; Fairbanks, H.; Bal, G.; Beer, E.; Tari, S. The Hubness Phenomenon in High-Dimensional Spaces. In *Research in Data Sciences*; Gasparovic, E., Domeniconi, C., Eds.; Association for Women in Mathematics; Springer: Cham, Switzerland, 2019; Volume 17, pp. 15–45.
41. Steinbach, M.; Ertöz, L.; Kumar, V. The Challenges of Clustering High Dimensional Data. In *New Directions in Statistics Physics*; Wille, L.T., Ed.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 273–309.
42. Domeniconi, C.; Gunopulos, D.; Ma, S.; Yan, B.; Al-Razgan, M.; Papadopoulos, D. Locally adaptive metrics for clustering high dimensional data. *Data Min. Knowl. Discov.* **2007**, *14*, 63–97. [CrossRef]
43. Gmx Cluster-GROMACS 2018. GROMACS User Guide—Gmx Cluster. Available online: Http://manual.gromacs.org/documentation/2018/onlinehelp/gmx-cluster.html (accessed on 1 May 2020).
44. Daura, X.; Gademann, K.; Jaun, B.; Seebach, D.; van Gunsteren, W.F.; Mark, A.E. Peptide Folding: When Simulation Meets Experiment. *Angew. Chem. Int. Ed.* **1999**, *38*, 236–240. [CrossRef]
45. Zhao, Q.; Hautamaki, V.; Fränti, P. Knee point detection in BIC for detecting the number of clusters. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems. Juan-les-Pins, France, 20–24 October 2008; pp. 664–673.
46. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [CrossRef]
47. McLachlan, G.J.; Basford, K.E. *Mixture Models: Inference and Applications to Clustering*; M Dekker: New York, NY, USA, 1988; Volume 84.
48. Geary, D. Mixture Models: Inference and Applications to Clustering. *J. R. Stat. Soc. Ser. A* **1989**, *152*, 126–127. [CrossRef]
49. Beeferman, D.; Berger, A. Agglomerative clustering of a search engine query log. *KDD* **2000**, *2000*, 407–416.
50. Gowda, K.C.; Krishna, G. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognit.* **1978**, *10*, 105–112. [CrossRef]
51. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
52. Zhang, G.J.; Zhou, X.G.; Yu, X.F.; Hao, H.; Yu, L. Enhancing protein conformational space sampling using distance profile-guided differential evolution. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2017**, *14*, 1288–1301. [CrossRef]
53. McLachlan, A.D. A mathematical procedure for superimposing atomic coordinates of proteins. *Acta Cryst. A* **1972**, *26*, 656–657. [CrossRef]